H 5

```
FFFFFFFFFFFFFFF         111               111        XXX         XXX
FFFFFFFFFFFFFFF         111               111        XXX         XXX
FFFFFFFFFFFFFFF         111               111        XXX         XXX
FFF                  111111            111111        XXX         XXX
FFF                  111111            111111        XXX         XXX
FFF                  111111            111111        XXX         XXX
FFF                     111               111           XXX   XXX
FFF                     111               111           XXX   XXX
FFF                     111               111           XXX   XXX
FFFFFFFFFFFF            111               111              XXX
FFFFFFFFFFFF            111               111              XXX
FFFFFFFFFFFF            111               111              XXX
FFF                     111               111           XXX   XXX
FFF                     111               111           XXX   XXX
FFF                     111               111           XXX   XXX
FFF                     111               111        XXX         XXX
FFF                     111               111        XXX         XXX
FFF                     111               111        XXX         XXX
FFF                  111111111         111111111     XXX         XXX
FFF                  111111111         111111111     XXX         XXX
FFF                  111111111         111111111     XXX         XXX
```

```
_$25

Symb
----
IOC$
IO_C
IO_C
IO_D
IO_F
IO_S
KILL

KILL
KILL
LB_E
LB_D
LB_F
LB_K
LB_L
LOCA
LOCA
LOCK

LOCK
LOCK
LOCK
LOC_
LOC_
L_CO
L_CO
L_DA
L_DA
MAIN
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE

MAKE
MAKE
MAP_
MAP_

MAP
MARK
MARK
MARK
MARK
MARK
```

```
**FILE**ID**QUOTAUTIL




    QQQQQQ    UU      UU    OOOOOO    TTTTTTTTTT    AAAAAA    UU      UU  TTTTTTTTTT    IIIIII    LL
    QQQQQQ    UU      UU    OOOOOO    TTTTTTTTTT    AAAAAA    UU      UU  TTTTTTTTTT    IIIIII    LL
  QQ      QQ  UU      UU  OO      OO      TT      AA      AA  UU      UU      TT          II      LL
  QQ      QQ  UU      UU  OO      OO      TT      AA      AA  UU      UU      TT          II      LL
  QQ      QQ  UU      UU  OO      OO      TT      AA      AA  UU      UU      TT          II      LL
  QQ      QQ  UU      UU  OO      OO      TT      AA      AA  UU      UU      TT          II      LL
  QQ      QQ  UU      UU  OO      OO      TT      AA      AA  UU      UU      TT          II      LL
  QQ      QQ  UU      UU  OO      OO      TT      AA      AA  UU      UU      TT          II      LL
  QQ  QQ  QQ  UU      UU  OO      OO      TT      AAAAAAAAAA  UU      UU      TT          II      LL
  QQ  QQ  QQ  UU      UU  OO      OO      TT      AAAAAAAAAA  UU      UU      TT          II      LL
  QQ    QQ    UU      UU  OO      OO      TT      AA      AA  UU      UU      TT          II      LL              ....
  QQ    QQ    UU      UU  OO      OO      TT      AA      AA  UU      UU      TT          II      LL              ....
    QQQQ  QQ  UUUUUUUUUU    OOOOOO        TT      AA      AA  UUUUUUUUUU      TT        IIIIII    LLLLLLLLLL      ....
    QQQQ  QQ  UUUUUUUUUU    OOOOOO        TT      AA      AA  UUUUUUUUUU      TT        IIIIII    LLLLLLLLLL      ....

  LL              IIIIII      SSSSSSSS
  LL              IIIIII      SSSSSSSS
  LL                II      SS
  LL                II      SS
  LL                II      SS
  LL                II      SS
  LL                II        SSSSSS
  LL                II        SSSSSS
  LL                II                SS
  LL                II                SS
  LL                II                SS
  LL                II                SS
  LLLLLLLLLL      IIIIII    SSSSSSSS
  LLLLLLLLLL      IIIIII    SSSSSSSS
```

```
0 MODULE QUOTAUTIL (
0                 LANGUAGE (BLISS32),
0                 IDENT = 'V04-001'
0                 ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 !
1 ! FACILITY:  F11ACP Structure Level 2
1 !
1 ! ABSTRACT:
1 !
1 !       This module contains routines that inplement the ACP control
1 !       functions that operate on the quota file.
1 !
1 ! ENVIRONMENT:
1 !
1 !       STARLET operating system, including privileged system services
1 !       and internal exec routines.
1 !
1 !--
1 !
1 !
1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  31-May-1979  15:18
1 !
1 ! MODIFIED BY:
1 !
1 !       V04-001 ACG0466         Andrew C. Goldstein,    12-Sep-1984  14:38
1 !               Flush quota file blocks from cache when disabling quotas
1 !
1 !       V03-012 CDS0008         Christian D. Saether    29-Aug-1984
1 !               Deal with potential multi-header quota file caused
1 !               by ACL's.
```

```
 1 !
 1 !       V03-011 CDS0007          Christian D. Saether    23-Aug-1984
 1 !               Mark quota fcb stale clusterwide when it is extended.
 1 !
 1 !       V03-010 ACG0438          Andrew C. Goldstein,    18-Jul-1984  20:32
 1 !               Implement quota cache lock; dequeue when cache is released.
 1 !               Use central dequeue routine.
 1 !
 1 !       V03-009 CDS0006          Christian D. Saether     9-May-1984
 1 !               Add serialization call to flush_quo_cache routine.
 1 !
 1 !       V03-008 CDS0005          Christian D. Saether    19-Apr-1984
 1 !               Bump REFCNT in quota file fcb also.
 1 !
 1 !       V03-007 ACG0412          Andrew C. Goldstein,    22-Mar-1984  18:35
 1 !               Implement agent access mode support; add access mode to
 1 !               protection check call
 1 !
 1 !       V03-006 ACG0400          Andrew C. Goldstein,     7-Mar-1984  17:07
 1 !               Implement cluster-wide quota cache, remove marking
 1 !               of SCB for quotas.
 1 !
 1 !       V03-005 CDS0004          Christian D. Saether     1-Mar-1984
 1 !               Remove reference to FLUSH_FID.
 1 !
 1 !       V03-004 CDS0003          Christian D. Saether    30-Dec-1983
 1 !               Use L_NORM linkage and BIND_COMMON macro.
 1 !
 1 !       V03-003 CDS0002          Christian D. Saether     6-Dec-1983
 1 !               Volume lock check on quota file modification request
 1 !               has changed.  NOALLOC is no longer set.
 1 !
 1 !       V03-002 CDS0001          Christian D. Saether    17-Oct-1983
 1 !               Add minimal quota checking support for xqp.
 1 !
 1 !       V03-001 ACG0308          Andrew C. Goldstein,    14-Jan-1983  14:26
 1 !               Fix consistency problems in linking FCB's
 1 !
 1 !       V02-005 ACG0213          Andrew C. Goldstein,    13-Aug-1981  13:42
 1 !               Remove write lock from quota file
 1 !
 1 !       V02-004 ACG0167          Andrew C. Goldstein,    16-Apr-1980  19:27
 1 !               Previous revision history moved to F11B.REV
 1 !**
 1
 1
 1 LIBRARY 'SYS$LIBRARY:LIB.L32';
 1 REQUIRE 'SRC$:FCPDEF.B32';
 1
 1
 1 FORWARD ROUTINE
 1         QUOTA_FILE_OP   : L_NORM NOVALUE, ! general quota file operations
 1         FLUSH_QUO_CACHE : L_NORM NOVALUE, ! flush dirty entries from quota cache
 1         DEACC_QFILE     : L_NORM,         ! deaccess the quota file
 1         RET_QENTRY      : L_NORM,         ! return quota file entry to user
 1         CONN_QFILE      : L_NORM NOVALUE, ! connect the quota file
 1         MAKE_QFCB       : L_NORM;         ! complete quota file access
```

```
1 GLOBAL ROUTINE QUOTA_FILE_OP (ABD, FIB) : L_NORM NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine implements most of the quota file ACP control functions
1 !       (i.e., the ones that are performed on the open quota file).
1 !
1 ! CALLING SEQUENCE:
1 !       QUOTA_FILE_OP (ARG1, ARG2)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: address of buffer descriptor packet
1 !       ARG2: address of user FIB
1 !
1 ! IMPLICIT INPUTS:
1 !       CLEANUP_FLAGS: cleanup action and status flags
1 !       CURRENT_VCB: VCB of current volume
1 !       IO_PACKET: I/O packet being processed
1 !       QUOTA_RECORD: record number of found quota file record
1 !       FREE_QUOTA: record number of first free quota file record
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       PRIMARY_FCB: FCB of quota file
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       quota file searched, modified, etc.
1 !
1 !--
1
2 BEGIN
2
2 MAP
2       ABD             : REF BBLOCKVECTOR [,ABD$C_LENGTH],
2                                       ! buffer descriptor vector
2       FIB             : REF BBLOCK;   ! user FIB
2
2 LITERAL
2       RECS_PER_BLOCK  = 512 / DQF$C_LENGTH,
2
2       MAX_QFUNC       = MAXU (FIB$C_DSA_QUOTA,
2                               FIB$C_EXA_QUOTA,
2                               FIB$C_REM_QUOTA,
2                               FIB$C_MOD_QUOTA,
2                               FIB$C_ADD_QUOTA
2                               ),
2
2       MIN_QFUNC       = MINU (FIB$C_DSA_QUOTA,
2                               FIB$C_EXA_QUOTA,
2                               FIB$C_REM_QUOTA,
                                FIB$C_MOD_QUOTA,
                                FIB$C_ADD_QUOTA
                                );

LOCAL
        TEMP1,                                  ! random temp storage
        TEMP2,                                  ! more of the same
        FCB             : REF BBLOCK,           ! address of quota file FCB
        BUFFER          : REF BBLOCK,           ! disk block buffer
        Q_RECORD        : REF BBLOCK,           ! record found in quota file
        Q_BLOCK         : REF BBLOCK;           ! quota arg block from user

BIND_COMMON;

EXTERNAL ROUTINE
        MAKE_FCB_STALE  : L_NORM NOVALUE,       ! mark fcb stale clusterwide
        SERIAL_FILE     : L_NORM,               ! serialize on given file
        ALLOCATION_LOCK : L_NORM,               ! serialize on volume allocation
        SWITCH_VOLUME   : L_NORM,               ! switch volume context
        SEARCH_QUOTA    : L_NORM,               ! find entry in quota file
        CHECK_PROTECT   : L_NORM,               ! check file protection
        GET_QUOTA_LOCK  : L_NORM,               ! take lock on quota cache entry
        REL_QUOTA_LOCK  : L_NORM,               ! release lock on quota cache entry
        WRITE_DIRTY     : L_NORM NOVALUE,       ! write dirty buffers
        READ_BLOCK      : L_NORM,               ! read a disk block
        EXTEND_CONTIG   : L_NORM,               ! extend a contiguous file
        WRITE_QUOTA     : L_NORM;               ! write quota file record


! Do the preliminary setup and validation. All operations handled by this
! routine operate on RVN 1 of a volume set and require the quota file to
! be connected.
!

SWITCH_VOLUME (1);
PRIMARY_FCB = FCB = .CURRENT_VCB[VCB$L_QUOTAFCB];
IF .FCB EQL 0
THEN ERR_EXIT (SS$_QFNOTACT);

SERIAL_FILE (FCB [FCB$W_FID]);

ALLOCATION_LOCK ();

! Do additional validation which is common for several functions. All but
! the disable function require a quota file search and require the quota
! argument block (P2) to be present.
!

IF .FIB[FIB$W_CNTRLFUNC] NEQ FIB$C_DSA_QUOTA
THEN
    BEGIN
    IF .ABD[ABD$C_NAME, ABD$W_COUNT] LSSU DQF$C_LENGTH
    THEN ERR_EXIT (SS$_INSFARG);
    Q_BLOCK = ABD[ABD$C_NAME, ABD$W_TEXT] + .ABD[ABD$C_NAME, ABD$W_TEXT] + 1;

    Q_RECORD = SEARCH_QUOTA (.Q_BLOCK[DQF$L_UIC], .FIB[FIB$L_CNTRLVAL], .FIB[FIB$L_WCC], 0);
    IF .FIB[FIB$V_ALL_MEM]
```

```
3           OR .FIB[FIB$V_ALL_GRP]
3           THEN FIB[FIB$L_WCC] = .QUOTA_RECORD;
3
3   ! All functions except disable and examine require write access to the
3   ! quota file; examine requires read access except when examining one's
3   ! own quota.
3   !
3
3       IF .FIB[FIB$W_CNTRLFUNC] NEQ FIB$C_EXA_QUOTA
3       THEN
3           CHECK_PROTECT (WRITE_ACCESS, 0, .FCB, 0)
3       ELSE
4           BEGIN
4           IF .FIB[FIB$V_ALL_MEM]
4           OR .FIB[FIB$V_ALL_GRP]
4           OR .Q_BLOCK[DQF$L_UIC] NEQ
4               .BBLOCK [.IO_PACKET[IRP$L_ARB], ARB$L_UIC]
4           THEN CHECK_PROTECT (READ_ACCESS, 0, .FCB, 0);
3           END;
3
3   ! All functions except disable and add require the quota file search to be
3   ! successful.
3   !
3
3       IF .FIB[FIB$W_CNTRLFUNC] NEQ FIB$C_ADD_QUOTA
3       THEN
3           IF .Q_RECORD EQL 0
3           THEN ERR_EXIT (SS$_NODISKQUOTA);
2       END;
2
2   ! Dispatch on the function and do it.
2   !
2
2   CASE .FIB[FIB$W_CNTRLFUNC] FROM MIN_QFUNC TO MAX_QFUNC OF
2       SET
2
2       [FIB$C_DSA_QUOTA]:                      ! disable disk quotas
3           BEGIN
3           IF NOT .CLEANUP_FLAGS[CLF_SYSPRV]
3           THEN ERR_EXIT (SS$_NOPRIV);
3           FLUSH_QUO_CACHE ();
3           WRITE_DIRTY (-1);
3           KERNEL_CALL (DEACC_QFILE);
2           END;
2
2       [FIB$C_EXA_QUOTA]:                      ! examine quota file entry
3           BEGIN
3           KERNEL_CALL (RET_QENTRY, .Q_RECORD, .ABD);
2           END;
2
2       [FIB$C_REM_QUOTA]:                      ! remove quota file entry
3           BEGIN
3           IF .Q_RECORD[DQF$L_USAGE] NEQ 0
3           THEN ERR_STATUS (SS$_OVRDSKQUOTA);
3           KERNEL_CALL (RET_QENTRY, .Q_RECORD, .ABD);
3           GET_QUOTA_LOCK (.QUOTA_INDEX, LCK$K_EXMODE);
3           CH$FILL (0, DQF$C_LENGTH, .Q_RECORD);
```

```
3               WRITE_QUOTA (.Q_RECORD);
3               REL_QUOTA_LOCK (.QUOTA_INDEX);
2               END;
2
2           [FIB$C_MOD_QUOTA]:                          ! modify quota file entry
3               BEGIN
3               IF .FIB[FIB$V_MOD_USE]
3               THEN
4                   BEGIN
4                   IF .BLOCK_LOCKID EQL 0
4                   THEN ERR_EXIT (SS$_ACCONFLICT);
4                   Q_RECORD[DQF$L_USAGE] = .Q_BLOCK[DQF$L_USAGE];
3                   END;
3               IF .FIB[FIB$V_MOD_PERM]
3               THEN
3                   Q_RECORD[DQF$L_PERMQUOTA] = .Q_BLOCK[DQF$L_PERMQUOTA];
3               IF .FIB[FIB$V_MOD_OVER]
3               THEN
3                   Q_RECORD[DQF$L_OVERDRAFT] = .Q_BLOCK[DQF$L_OVERDRAFT];
3               IF .Q_RECORD[DQF$L_USAGE] GTRU .Q_RECORD[DQF$L_PERMQUOTA]
3               THEN ERR_STATUS (SS$_OVRDSKQUOTA);
3               WRITE_QUOTA (.Q_RECORD);
3               KERNEL_CALL (RET_QENTRY, .Q_RECORD, .ABD);
2               END;
2
2           [FIB$C_ADD_QUOTA]:                          ! add quota file entry
3               BEGIN
3               IF .Q_RECORD NEQ 0
3               THEN ERR_EXIT (SS$_DUPDSKQUOTA);
3               IF .FREE_QUOTA EQL 0
3               THEN
4                   BEGIN
4                   IF .FCB[FCB$L_FILESIZE] GEQU (1^24)/RECS_PER_BLOCK-1
4                   THEN ERR_EXIT (SS$_DEVICEFULL);
4                   TEMP1 = .FIB[FIB$W_CNTRLFUNC];
4                   TEMP2 = .FIB[FIB$L_CNTRLVAL];
4                   Q_RECORD = EXTEND_CONTIG (.FIB, .FCB, 1);
4                   MAKE_FCB_STALE (.FCB);
4                   FIB[FIB$W_CNTRLFUNC] = .TEMP1;
4                   FIB[FIB$L_CNTRLVAL] = .TEMP2;
4                   FIB[FIB$L_EXVBN] = 0;
4                   END
3               ELSE
4                   BEGIN
4                   Q_RECORD = READ_BLOCK ((.FREE_QUOTA-1)/RECS_PER_BLOCK + .FCB[FCB$L_STLBN],
4                                           1, QUOTA_TYPE);
4                   Q_RECORD = .Q_RECORD + ((.FREE_QUOTA-1) MOD RECS_PER_BLOCK) * DQF$C_LENGTH;
3                   END;
3
3               CH$FILL (0, DQF$C_LENGTH, .Q_RECORD);
3               Q_RECORD[DQF$V_ACTIVE] = 1;
3               Q_RECORD[DQF$L_UIC] = .Q_BLOCK[DQF$L_UIC];
3               Q_RECORD[DQF$L_USAGE] = .Q_BLOCK[DQF$L_USAGE];
3               Q_RECORD[DQF$L_PERMQUOTA] = .Q_BLOCK[DQF$L_PERMQUOTA];
3               Q_RECORD[DQF$L_OVERDRAFT] = .Q_BLOCK[DQF$L_OVERDRAFT];
3               WRITE_QUOTA (.Q_RECORD);
2               END;
```

```
;  344        1333 2
;  345        1334 2      [INRANGE, OUTRANGE]:         0;        ! should not be called with other functions
;  346        1335 2
;  347        1336 2      TES;
;  348        1337 2
;  349        1338 1 END;                                        ! end of routine QUOTA_FILE_OP


                                                                  .TITLE  QUOTAUTIL
                                                                  .IDENT  \V04-001\

                                                                  .EXTRN  MAKE_FCB_STALE, SERIAL_FILE
                                                                  .EXTRN  ALLOCATION_LOCK
                                                                  .EXTRN  SWITCH_VOLUME, SEARCH_QUOTA
                                                                  .EXTRN  CHECK_PROTECT, GET_QUOTA_LOCK
                                                                  .EXTRN  REL_QUOTA_LOCK, WRITE_DIRTY
                                                                  .EXTRN  READ_BLOCK, EXTEND_CONTIG
                                                                  .EXTRN  WRITE_QUOTA

                                                                  .PSECT  $CODE$,NOWRT,2

                                         03FC 00000               .ENTRY  QUOTA_FILE_OP, Save R2,R3,R4,R5,R6,R7,R8,R9  ; 1105
                     59      0000G   CF    9E 00002               MOVAB   WRITE_QUOTA, R9                               ;
                                     01    DD 00007               PUSHL   #1                                            ; 1196
                0000G   CF           01    FB 00009               CALLS   #1, SWITCH_VOLUME                             ;
                     50       98     AA    D0 0000E               MOVL    -104(BASE), R0                                ; 1197
                     58       54     A0    D0 00012               MOVL    84(R0), FCB                                   ;
                08   AA              58    D0 00016               MOVL    FCB, 8(BASE)                                  ;
                                     05    12 0001A               BNEQ    1$                                            ; 1198
                            03D4     8F    BF 0001C               CHMU    #980                                          ; 1199
                                           04 00020               RET                                                   ;
                     24              A8    9F 00021 1$:           PUSHAB  36(FCB)                                       ; 1201
                0000G   CF           01    FB 00024               CALLS   #1, SERIAL_FILE                               ;
                0000G   CF           00    FB 00029               CALLS   #0, ALLOCATION_LOCK                           ; 1203
                     50       08     AC    D0 0002E               MOVL    FIB, R0                                       ; 1210
                     0A       16     A0    B1 00032               CMPW    22(R0), #10                                   ;
                                     03    12 00036               BNEQ    2$                                            ;
                                   008F    31 00038               BRW     10$                                           ;
                     50       04     AC    D0 0003B 2$:           MOVL    ABD, R0                                       ; 1213
                     20       12     A0    B1 0003F               CMPW    18(R0), #32                                   ;
                                     05    1E 00043               BGEQU   3$                                            ;
                            0114     8F    BF 00045               CHMU    #276                                          ; 1214
                                           04 00049               RET                                                   ;
                     51       04     AC    D0 0004A 3$:           MOVL    ABD, R1                                       ; 1215
                     50       10     A1    3C 0004E               MOVZWL  16(R1), R0                                    ;
                     57       11   A140    9E 00052               MOVAB   17(R1)[R0], Q_BLOCK                           ;
                                     7E    D4 00057               CLRL    -(SP)                                         ; 1217
                     50       08     AC    D0 00059               MOVL    FIB, R0                                       ;
                              10     A0    DD 0005D               PUSHL   16(R0)                                        ;
                              18     A0    DD 00060               PUSHL   24(R0)                                        ;
                              04     A7    DD 00063               PUSHL   4(Q_BLOCK)                                    ;
                0000G   CF           04    FB 00066               CALLS   #4, SEARCH_QUOTA                              ;
                     56              50    D0 0006B               MOVL    R0, Q_RECORD                                  ;
                     50       08     AC    D0 0006E               MOVL    FIB, R0                                       ; 1218
                     05       18     A0    E8 00072               BLBS    24(R0), 4$                                    ;
          06         18  A0          01    E1 00076               BBC     #1, 24(R0), 5$                                ; 1219
                     10  A0   02B4   CA    D0 0007B 4$:           MOVL    692(BASE), 16(R0)                             ; 1220
```

```
                                                    L 16
QUOTAUTIL                                    16-Sep-1984 00:51:04    VAX-11 Bliss-32 V4.0-742            Page   8
V04-001                                      14-Sep-1984 12:30:41    [F11X.SRC]QUOTAUTIL.B32;2                 (2)

                            52       08   AC  D0 00081  5$:   MOVL    FIB, R2                          ; 1227
                            0C       16   A2  B1 00085        CMPW    22(R2), #12                      ;
                                          09  13 00089        BEQL    6$                               ;
                                          7E  D4 0008B        CLRL    -(SP)                            ; 1229
                                          58  DD 0008D        PUSHL   FCB                              ;
                            7E            01  7D 0008F        MOVQ    #1, -(SP)                        ;
                                          1E  11 00092        BRB     8$                               ;
                            14       18   A2  E8 00094  6$:   BLBS    24(R2), 7$                       ; 1232
                 0F     18  A2            01  E0 00098        BBS     #1, 24(R2), 7$                   ; 1233
                            50       90   AA  D0 0009D        MOVL    -112(BASE), R0                   ; 1235
                            50       58   A0  D0 000A1        MOVL    88(R0), R0                       ;
                        38  A0       04   A7  D1 000A5        CMPL    4(Q_BLOCK), 56(R0)               ;
                                          0B  13 000AA        BEQL    9$                               ;
                                          7E  D4 000AC  7$:   CLRL    -(SP)                            ; 1236
                                          58  DD 000AE        PUSHL   FCB                              ;
                                          7E  7C 000B0        CLRQ    -(SP)                            ;
                     0000G  CF            04  FB 000B2  8$:   CALLS   #4, CHECK_PROTECT                ;
                            50       08   AC  D0 000B7  9$:   MOVL    FIB, R0                          ; 1243
                            0B       16   A0  B1 000BB        CMPW    22(R0), #11                      ;
                                          09  13 000BF        BEQL    10$                              ;
                                          56  D5 000C1        TSTL    Q_RECORD                         ; 1245
                                          05  12 000C3        BNEQ    10$                              ;
                                   03E4   8F  BF 000C5        CHMU    #996                             ; 1246
                                              04 000C9        RET                                      ;
                            50       08   AC  D0 000CA  10$:  MOVL    FIB, R0                          ; 1252
                 04         0A       16   A0  AF 000CE        CASEW   22(R0), #10, #4                  ;
     005E      00A5       00B0          000B     000D3  11$:  .WORD   12$-11$,-                        ;
                                        0025     000DB                23$-11$,-                        ;
                                                                      22$-11$,-                        ;
                                                                      16$-11$,-                        ;
                                                                      14$-11$                          ;
                                              04 000DD        RET                                      ;
                            03       01   AA  E8 000DE  12$:  BLBS    1(BASE), 13$                     ; 1257
                                          24  BF 000E2        CHMU    #36                              ; 1258
                                              04 000E4        RET                                      ;
                     0000V  CF            00  FB 000E5  13$:  CALLS   #0, FLUSH_QUO_CACHE              ; 1259
                            7E            01  CE 000EA        MNEGL   #1, -(SP)                        ; 1260
                     0000G  CF            01  FB 000ED        CALLS   #1, WRITE_DIRTY                  ;
                     0000V  CF            00  FB 000F2        CALLS   #0, DEACC_QFILE                  ; 1261
                                              04 000F7        RET                                      ; 1252
                                     08   A6  D5 000F8  14$:  TSTL    8(Q_RECORD)                      ; 1271
                                          0A  13 000FB        BEQL    15$                              ;
                            06       80   AA  E9 000FD        BLBC    -128(BASE), 15$                  ; 1272
                        80  AA     0669   8F  B0 00101        MOVW    #1641, -128(BASE)                ;
                                     04   AC  DD 00107  15$:  PUSHL   ABD                              ; 1273
                                          56  DD 0010A        PUSHL   Q_RECORD                         ;
                     0000V  CF            02  FB 0010C        CALLS   #2, RET_QENTRY                   ;
                                          05  DD 00111        PUSHL   #5                               ; 1274
                                   02C0   CA  DD 00113        PUSHL   704(BASE)                        ;
                     0000G  CF            02  FB 00117        CALLS   #2, GET_QUOTA_LOCK               ;
       20        00         6E            00  2C 0011C        MOVC5   #0, (SP), #0, #32, (Q_RECORD)    ; 1275
                                          66     00121                                                 ;
                                          56  DD 00122        PUSHL   Q_RECORD                         ; 1276
                            69            01  FB 00124        CALLS   #1, WRITE_QUOTA                  ;
                                   02C0   CA  DD 00127        PUSHL   704(BASE)                        ; 1277
                     0000G  CF            01  FB 0012B        CALLS   #1, REL_QUOTA_LOCK               ;
                                              04 00130        RET                                      ; 1252
```

```
                   10        18   A0            02  E1 00131 16$:      BBC      #2, 24(R0), 18$                                   ; 1282
                                        FF7C    CA  D5 00136           TSTL     -132(BASE)                                        ; 1285
                                                05  12 0013A           BNEQ     17$                                               ;
                                        0800    8F  BF 0013C           CHMU     #2048                                             ; 1286
                                                    04 00140           RET                                                        ;
                             08   A6      08    A7  D0 00141 17$:      MOVL     8(Q_BLOCK), 8(Q_RECORD)                           ; 1287
                                  50      08    AC  D0 00146 18$:      MOVL     FIB, R0                                           ; 1289
                   05        18   A0            03  E1 0014A           BBC      #3, 24(R0), 19$                                   ;
                             0C   A6      0C    A7  D0 0014F           MOVL     12(Q_BLOCK), 12(Q_RECORD)                         ; 1291
                                  50      08    AC  D0 00154 19$:      MOVL     FIB, R0                                           ; 1292
                   05        18   A0            04  E1 00158           BBC      #4, 24(R0), 20$                                   ;
                             10   A6      10    A7  D0 0015D           MOVL     16(Q_BLOCK), 16(Q_RECORD)                         ; 1294
                             0C   A6      08    A6  D1 00162 20$:      CMPL     8(Q_RECORD), 12(Q_RECORD)                         ; 1295
                                                0A  1B 00167           BLEQU    21$                                               ;
                             06   80            AA  E9 00169           BLBC     -128(BASE), 21$                                   ; 1296
                             80   AA    0669    8F  B0 0016D           MOVW     #1641, -128(BASE)                                 ;
                                                56  DD 00173 21$:      PUSHL    Q_RECORD                                          ; 1297
                                  69            01  FB 00175           CALLS    #1, WRITE_QUOTA                                   ;
                                          04    AC  DD 00178 22$:      PUSHL    ABD                                               ; 1298
                                                56  DD 0017B           PUSHL    Q_RECORD                                          ;
                        0000V     CF            02  FB 0017D           CALLS    #2, RET_QENTRY                                    ;
                                                    04 00182           RET                                                        ; 1252
                                                56  D5 00183 23$:      TSTL     Q_RECORD                                          ; 1303
                                                05  13 00185           BEQL     24$                                               ;
                                        03DC    8F  BF 00187           CHMU     #988                                              ; 1304
                                                    04 0018B           RET                                                        ;
                                  50    02B8    CA  D0 0018C 24$:      MOVL     696(BASE), R0                                     ; 1305
                                                49  12 00191           BNEQ     26$                                               ;
                     000FFFFF     8F      38    A8  D1 00193           CMPL     56(FCB), #1048575                                 ; 1308
                                                05  1F 0019B           BLSSU    25$                                               ;
                                        0850    8F  BF 0019D           CHMU     #2128                                             ; 1309
                                                    04 001A1           RET                                                        ;
                                  50      08    AC  D0 001A2 25$:      MOVL     FIB, R0                                           ; 1310
                                  53      16    A0  3C 001A6           MOVZWL   22(R0), TEMP1                                     ;
                                  52      18    A0  D0 001AA           MOVL     24(R0), TEMP2                                     ; 1311
                                                01  DD 001AE           PUSHL    #1                                                ; 1312
                                        0101    8F  BB 001B0           PUSHR    #^M<R0,R8>                                        ;
                        0000G     CF            03  FB 001B4           CALLS    #3, EXTEND_CONTIG                                 ;
                                  56            50  D0 001B9           MOVL     R0, Q_RECORD                                      ;
                                                58  DD 001BC           PUSHL    FCB                                               ; 1313
                        0000G     CF            01  FB 001BE           CALLS    #1, MAKE_FCB_STALE                                ;
                                  50      08    AC  D0 001C3           MOVL     FIB, R0                                           ; 1314
                             16   A0            53  B0 001C7           MOVW     TEMP1, 22(R0)                                     ;
                                  50      08    AC  D0 001CB           MOVL     FIB, R0                                           ; 1315
                             18   A0            52  D0 001CF           MOVL     TEMP2, 24(R0)                                     ;
                                  50      08    AC  D0 001D3           MOVL     FIB, R0                                           ; 1316
                                          1C    A0  D4 001D7           CLRL     28(R0)                                            ;
                                                2B  11 001DA           BRB      27$                                               ; 1305
                                                05  DD 001DC 26$:      PUSHL    #5                                                ; 1320
                                                01  DD 001DE           PUSHL    #1                                                ;
                                                50  D7 001E0           DECL     R0                                                ;
                                  50            10  C6 001E2           DIVL2    #16, R0                                           ;
                                       30 B840  9F 001E5               PUSHAB   @48(FCB)[R0]                                      ;
                        0000G     CF            03  FB 001E9           CALLS    #3, READ_BLOCK                                    ;
                                  56            50  D0 001EE           MOVL     R0, Q_RECORD                                      ;
       7E FFFFFFFF       8F     02B8    CA      01  7A 001F1           EMUL     #1, 696(BASE), #-1, -(SP)                         ; 1322
       50                50             8E      10  7B 001FC           EDIV     #16, (SP)+, R0, R0                                ;
```

```
                                        50          20  C4 00201          MULL2   #32, R0                                        ;
                                        56          50  C0 00204          ADDL2   R0, Q_RECORD                                   ;
           20              00           6E          00  2C 00207 27$:     MOVC5   #0, (SP), #0, #32, (Q_RECORD)                  ; 1325
                                                    66     0020C                                                                 ;
                                        66          01  88 0020D          BISB2   #1, (Q_RECORD)                                 ; 1326
                                  04    A6    04    A7  7D 00210          MOVQ    4(Q_BLOCK), 4(Q_RECORD)                        ; 1327
                                  0C    A6    0C    A7  7D 00215          MOVQ    12(Q_BLOCK), 12(Q_RECORD)                      ; 1329
                                                    56  DD 0021A          PUSHL   Q_RECORD                                       ; 1331
                                        69          01  FB 0021C          CALLS   #1, WRITE_QUOTA                                ;
                                                        04 0021F          RET                                                    ; 1338

; Routine Size:  544 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 GLOBAL ROUTINE FLUSH_QUO_CACHE : L_NORM NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine flushes dirty entries in the quota cache back to the
1 !       quota file.
1 !
1 !
1 ! CALLING SEQUENCE:
1 !       FLUSH_QUO_CACHE ()
1 !
1 ! INPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !       CURRENT_VCB: VCB of volume
1 !       context set to RVN 1
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       quota cache flushed, quota file modified
1 !
1 !--
1
2 BEGIN
2
2 BUILTIN
2       FP;
2
2 LITERAL
2       RECS_PER_BLOCK  = 512 / DQF$C_LENGTH;
2
2 LOCAL
2       QUOTA_CACHE     : REF BBLOCK,   ! address of quota cache
2       QUOTA_LIST      : REF BBLOCKVECTOR [,VCA$C_QUOLENGTH],
2                                       ! address of cache entries
2       FCB             : REF BBLOCK,   ! address of quota file FCB
2       REC_NUM,                        ! record number to read
2       STATUS,                         ! system service status
2       Q_RECORD        : REF BBLOCK,   ! address of record read
2       LOCK_STATUS     : VECTOR [2];   ! LKSB for lock conversion
2
2
2 BIND_COMMON;
2
2 EXTERNAL ROUTINE
2       ZERO_ON_ERROR,                  ! return zero on error signal (handler)
```

```
    2           ALLOCATION_LOCK : L_NORM NOVALUE, ! serialize on volume
    2           READ_BLOCK      : L_NORM,         ! read a disk block
    2           CLEAN_QUO_CACHE : L_NORM,         ! flush cache entry to record
    2           REL_QUOTA_LOCK  : L_NORM;         ! release lock on cache entry
    2
    2
    2   ! Set up the condition handler to handle I/O errors.
    2   !
    2
    2   .FP = ZERO_ON_ERROR;
    2
    2   ! Scan the quota cache, looking for valid dirty entries. If one is found,
    2   ! read its record from the quota file, update the record, and write it back.
    2   !
    2
    2   QUOTA_CACHE = .CURRENT_VCB[VCB$L_QUOCACHE];
    2   IF .QUOTA_CACHE EQL 0 THEN RETURN;        ! nop if no quota cache
    2
    2   ALLOCATION_LOCK ();
    2
    2   FCB = .CURRENT_VCB[VCB$L_QUOTAFCB];
    2   QUOTA_LIST = QUOTA_CACHE[VCA$L_QUOLIST];
    2   INCR J FROM 1 TO .QUOTA_CACHE[VCA$W_QUOSIZE]
    2   DO
    3       BEGIN
    3       IF  .QUOTA_LIST[.J-1, VCA$V_QUODIRTY]
    3       AND .QUOTA_LIST[.J-1, VCA$L_QUORECNUM] NEQ 0
    3       THEN
    4           BEGIN
    4           REC_NUM = .QUOTA_LIST[.J-1, VCA$L_QUORECNUM] - 1;
    4           Q_RECORD = READ_BLOCK (.REC_NUM / RECS_PER_BLOCK
    4                                  + .FCB[FCB$L_ST[BN], 1, QUOTA_TYPE)
    4                      + (.REC_NUM MOD RECS_PER_BLOCK) * DQF$C_LENGTH;
    4           IF .Q_RECORD GEQA 512
    4           THEN KERNEL_CALL (CLEAN_QUO_CACHE, .J, .Q_RECORD);
    3           END;
    3       REL_QUOTA_LOCK (.J);
    2       END;
    2
    2   ! Now mark the quota cache invalid. If we are holding a cache lock,
    2   ! demote it down to NL to indicate that we are no longer holding
    2   ! cache contents.
    2   !
    2
    2   QUOTA_CACHE[VCA$V_CACHEVALID] = 0;
    2   IF .QUOTA_CACHE[VCA$L_QUOCLKID] NEQ 0
    2   THEN
    3       BEGIN
    3       LOCK_STATUS[1] = .QUOTA_CACHE[VCA$L_QUOCLKID];
P   3       STATUS = $ENQW (EFN    = EFN,
P   3                       LKMODE = LCK$K_NLMODE,
P   3                       FLAGS  = LCK$M_NOQUEUE OR LCK$M_SYNCSTS OR LCK$M_CVTSYS OR LCK$M_CONVERT,
P   3                       LKSB   = LOCK_STATUS
    3                       );
    3       IF NOT .STATUS
    3       THEN BUG_CHECK (XQPERR, FATAL, 'Unexpected lock manager error');
    2       END;
```

```
QUOTAUTIL                                                  E 1
V04-001                                                    16-Sep-1984 00:51:04    VAX-11 Bliss-32 V4.0-742          Page  13
                                                           14-Sep-1984 12:30:41    [F11X.SRC]QUOTAUTIL.B32;2               (3)

;  465           1453  2
;  466           1454  1 END;                              ! end of routine FLUSH_QUO_CACHE


                                                                    .EXTRN  ZERO_ON_ERROR, CLEAN_QUO_CACHE
                                                                    .EXTRN  SYS$ENQW, BUG$_XQPERR

                                              01FC 00000            .ENTRY  FLUSH_QUO_CACHE, Save R2,R3,R4,R5,R6,R7,R8   ; 1339
                         5E              08    C2 00002             SUBL2   #8, SP                                       ;
                         6D       0000G  CF    9E 00005             MOVAB   ZERO_ON_ERROR, (FP)                          ; 1405
                         50          98  AA    D0 0000A             MOVL    -104(BASE), R0                               ; 1411
                         52          5C  A0    D0 0000E             MOVL    92(R0), QUOTA_CACHE                          ;
                                         7A    13 00012             BEQL    4$                                           ; 1412
                 0000G   CF              00    FB 00014             CALLS   #0, ALLOCATION_LOCK                          ; 1414
                         50          98  AA    D0 00019             MOVL    -104(BASE), R0                               ; 1416
                         56          54  A0    D0 0001D             MOVL    84(R0), FCB                                  ;
                         53          44  A2    9E 00021             MOVAB   68(R2), QUOTA_LIST                           ; 1417
                         58              62    3C 00025             MOVZWL  (QUOTA_CACHE), R8                            ; 1418
                                         54    D4 00028             CLRL    J                                            ;
                                         57    11 0002A             BRB     3$                                           ;
              50         54              1C    C5 0002C  1$:        MULL3   #28, J, R0                                   ; 1421
                         50              53    C0 00030             ADDL2   QUOTA_LIST, R0                               ;
              44    EF   A0              01    E1 00033             BBC     #1, -17(R0), 2$                              ;
   00       EC  A0       18              00    ED 00038             CMPZV   #0, #24, -20(R0), #0                         ; 1422
                                         3C    13 0003E             BEQL    2$                                           ;
   55       EC  A0       18              00    EF 00040             EXTZV   #0, #24, -20(R0), REC_NUM                    ; 1425
                                         55    D7 00046             DECL    REC_NUM                                      ;
                                         05    DD 00048             PUSHL   #5                                           ; 1426
                                         01    DD 0004A             PUSHL   #1                                           ;
              50         55              10    C7 0004C             DIVL3   #16, REC_NUM, R0                             ;
                                   30  B640    9F 00050             PUSHAB  @48(FCB)[R0]                                 ; 1427
                 0000G   CF              03    FB 00054             CALLS   #3, READ_BLOCK                               ;
   7E         00         55              01    7A 00059             EMUL    #1, REC_NUM, #0, -(SP)                       ; 1428
   51         51         8E              10    7B 0005E             EDIV    #16, (SP)+, R1, R1                           ;
                         51              20    C4 00063             MULL2   #32, R1                                      ;
              57         50              51    C1 00066             ADDL3   R1, R0, Q_RECORD                             ;
          00000200       8F              57    D1 0006A             CMPL    Q_RECORD, #512                               ; 1429
                                         09    1F 00071             BLSSU   2$                                           ;
                                0090     8F    BB 00073             PUSHR   #^M<R4,R7>                                   ; 1430
                 0000G   CF              02    FB 00077             CALLS   #2, CLEAN_QUO_CACHE                          ;
                                         54    DD 0007C  2$:        PUSHL   J                                            ; 1432
                 0000G   CF              01    FB 0007E             CALLS   #1, REL_QUOTA_LOCK                           ;
              A5         54              58    F3 00083  3$:        AOBLEQ  R8, J, 1$                                    ; 1418
                    0B   A2              01    8A 00087             BICB2   #1, 11(QUOTA_CACHE)                          ; 1440
                                     04  A2    D5 0008B             TSTL    4(QUOTA_CACHE)                               ; 1441
                                         25    13 0008E  4$:        BEQL    5$                                           ;
                    04   AE          04  A2    D0 00090             MOVL    4(QUOTA_CACHE), LOCK_STATUS+4                ; 1444
                                         7E    7C 00095             CLRQ    -(SP)                                        ; 1449
                                         7E    7C 00097             CLRQ    -(SP)                                        ;
                                         7E    7C 00099             CLRQ    -(SP)                                        ;
                                         7E    D4 0009B             CLRL    -(SP)                                        ;
                         7E          4E  8F    9A 0009D             MOVZBL  #78, -(SP)                                   ;
                                     20  AE    9F 000A1             PUSHAB  LOCK_STATUS                                  ;
                         7E              1E    7D 000A4             MOVQ    #30, -(SP)                                   ;
             00000000G   00              0B    FB 000A7             CALLS   #11, SYS$ENQW                                ;
                         04              50    E8 000AE             BLBS    STATUS, 5$                                   ; 1450
```

```
                                   FEFF 000B1          BUGW                                        ; 1451
                                  0000* 000B3          .WORD     <BUG$_XQPERR!4>                   ;
                                     04 000B5 5$:      RET                                         ; 1454

; Routine Size:  182 bytes,    Routine Base:  $CODE$ + 0220
```

```
1 GLOBAL ROUTINE DEACC_QFILE : L_NORM =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine deaccesses the quota file and releases the FCB if it
1 !       is idle. This routine must be aclled in kernel mode.
1 !
1 ! CALLING SEQUENCE:
1 !       DEACC_QFILE ()
1 !
1 ! INPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !       CURRENT_VCB: VCB of volume
1 !       context set to RVN 1
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       1
1 !
1 ! SIDE EFFECTS:
1 !       quota file disconnected from VCB, FCB deallocated
1 !
1 !--
1
2 BEGIN
2
2 LOCAL
2       ACCTL,                          ! calculate remaining access control
2       LCKMODE,                        ! lock mode to convert access lock to.
2       FCB             : REF BBLOCK,   ! FCB of quota file
2       STATUS,                         ! system service status
2       QUOTA_CACHE     : REF BBLOCK;   ! address of quota cache block
2
2 BIND_COMMON;
2
2 EXTERNAL ROUTINE
2       KILL_BUFFERS    : L_NORM,       ! flush specified buffers from cache
2       CONV_ACCLOCK    : L_NORM,       ! convert access lock
2       LOCK_MODE       : L_JSB_1ARG,   ! calculate lock mode from access ctl
2       DEQ_LOCK        : L_NORM,       ! dequeue a lock
2       DEALLOCATE      : L_NORM ADDRESSING_MODE (GENERAL); ! deallocate system dynamic memory
2
2
2 ! Flush the quota file data blocks from the block buffer cache.
2 !
2
2 KILL_BUFFERS (1, -1);
2
```

```
;  525          1512  2 ! Decrement access and lock counts on the FCB.
;  526          1513  2 !
;  527          1514  2
;  528          1515  2 PRIMARY_FCB = FCB = .CURRENT_VCB[VCB$L_QUOTAFCB];
;  529          1516  2 CURRENT_VCB[VCB$L_QUOTAFCB] = 0;
;  530          1517  2
;  531          1518  2 ACCTL = 0;
;  532          1519  2
;  533          1520  2 IF .FCB[FCB$W_WCNT] NEQ 0
;  534          1521  2 THEN ACCTL = FIB$M_WRITE;
;  535          1522  2
;  536          1523  2 FCB[FCB$W_TCNT] = .FCB[FCB$W_TCNT] - 1;
;  537          1524  2
;  538          1525  2 LCKMODE = 0;
;  539          1526  2
;  540          1527  2 IF (FCB[FCB$W_ACNT] = .FCB[FCB$W_ACNT] - 1) NEQ 0
;  541          1528  2 THEN
;  542          1529  2     LCKMODE = LOCK_MODE (.ACCTL);
;  543          1530  2
;  544          1531  2 FCB[FCB$W_REFCNT] = .FCB[FCB$W_REFCNT] - 1;
;  545          1532  2
;  546          1533  2 ! Convert the access lock to reflect the remaining accessors.
;  547          1534  2 !
;  548          1535  2
;  549          1536  2 CONV_ACCLOCK (.LCKMODE, .FCB);
;  550          1537  2
;  551          1538  2 ! Release the quota cache lock, if there was one. Unlink and deallocate
;  552          1539  2 ! the quota cache block.
;  553          1540  2 !
;  554          1541  2
;  555          1542  2 QUOTA_CACHE = .CURRENT_VCB[VCB$L_QUOCACHE];
;  556          1543  2 IF .QUOTA_CACHE[VCA$L_QUOCLKID] NEQ 0
;  557          1544  2 THEN
;  558          1545  3     BEGIN
;  559          1546  3     DEQ_LOCK (.QUOTA_CACHE[VCA$L_QUOCLKID]);
;  560          1547  2     END;
;  561          1548  2
;  562          1549  2 DEALLOCATE (.QUOTA_CACHE);
;  563          1550  2 CURRENT_VCB[VCB$L_QUOCACHE] = 0;
;  564          1551  2
;  565          1552  2 RETURN 1;
;  566          1553  2
;  567          1554  1 END;                                       ! end of routine DEACC_QFILE


                                                        .EXTRN  KILL_BUFFERS, CONV_ACCLOCK
                                                        .EXTRN  LOCK_MODE, DEQ_LOCK
                                                        .EXTRN  DEALLOCATE

                                      000C 00000        .ENTRY  DEACC_QFILE, Save R2,R3            ; 1455
                    7E             01   CE 00002        MNEGL   #1, -(SP)                          ; 1510
                                   01   DD 00005        PUSHL   #1                                 ;
            0000G   CF             02   FB 00007        CALLS   #2, KILL_BUFFERS                   ;
                    50       98    AA   D0 0000C        MOVL    -104(BASE), R0                     ; 1515
                    52       54    A0   D0 00010        MOVL    84(R0), FCB                        ;
               08   AA             52   D0 00014        MOVL    FCB, 8(BASE)                       ;
```

```
                  50       98  AA  D0 00018         MOVL    -104(BASE), R0                    ; 1516
                           54  A0  D4 0001C         CLRL    84(R0)                            ;
                               50  D4 0001F         CLRL    ACCTL                             ; 1518
                  1C           A2  B5 00021         TSTW    28(FCB)                           ; 1520
                               05  13 00024         BEQL    1$                                ;
                  50     0100  8F  3C 00026         MOVZWL  #256, ACCTL                       ; 1521
                           20  A2  B7 0002B 1$:     DECW    32(FCB)                           ; 1523
                               53  D4 0002E         CLRL    LCKMODE                           ; 1525
                  51       1A  A2  3C 00030         MOVZWL  26(FCB), R1                       ; 1527
                               51  D7 00034         DECL    R1                                ;
            1A    A2           51  B0 00036         MOVW    R1, 26(FCB)                       ;
                               51  D5 0003A         TSTL    R1                                ;
                               06  13 0003C         BEQL    2$                                ;
                           0000G  30 0003E         BSBW    LOCK_MODE                         ; 1529
                  53           50  D0 00041         MOVL    R0, LCKMODE                       ;
                           18  A2  B7 00044 2$:     DECW    24(FCB)                           ; 1531
                               52  DD 00047         PUSHL   FCB                               ; 1536
                               53  DD 00049         PUSHL   LCKMODE                           ;
           0000G  CF           02  FB 0004B         CALLS   #2, CONV_ACCLOCK                  ;
                  50       98  AA  D0 00050         MOVL    -104(BASE), R0                    ; 1542
                  52       5C  A0  D0 00054         MOVL    92(R0), QUOTA_CACHE               ;
                           04  A2  D5 00058         TSTL    4(QUOTA_CACHE)                    ; 1543
                               08  13 0005B         BEQL    3$                                ;
                           04  A2  DD 0005D         PUSHL   4(QUOTA_CACHE)                    ; 1546
           0000G  CF           01  FB 00060         CALLS   #1, DEQ_LOCK                      ;
                               52  DD 00065 3$:     PUSHL   QUOTA_CACHE                       ; 1549
       00000000G  00           01  FB 00067         CALLS   #1, DEALLOCATE                    ;
                  50       98  AA  D0 0006E         MOVL    -104(BASE), R0                    ; 1550
                           5C  A0  D4 00072         CLRL    92(R0)                            ;
                  50           01  D0 00075         MOVL    #1, R0                            ; 1552
                                   04 00078         RET                                       ; 1554

; Routine Size:  121 bytes,    Routine Base:  $CODE$ + 02D6
```

```
1 GLOBAL ROUTINE RET_QENTRY (Q_RECORD, ABD) : L_NORM =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine copies the specified quota file record into the
1 !       result string area of the buffer descriptor packet. This routine
1 !       must be called in kernel mode.
1 !
1 ! CALLING SEQUENCE:
1 !       RET_QENTRY (ARG1, ARG2)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: address of quota file record
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       ARG2: address of buffer descriptor packet
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       1
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2
2 MAP
2       Q_RECORD        : REF BBLOCK,   ! quota file record
2       ABD             : REF BBLOCKVECTOR [,ABD$C_LENGTH];
2                                       ! descriptor arg
2
2 ! If the user provided a result length buffer, give him the length
2 ! of the record.
2 !
2
2 IF .ABD[ABD$C_RESL, ABD$W_COUNT] GEQ 2
2 THEN
3     BEGIN
3     (.ABD[ABD$C_RESL, ABD$W_TEXT] + ABD[ABD$C_RESL, ABD$W_TEXT] + 1)<0,16> = DQF$C_LENGTH;
2     END;
2
2 ! If the user provided a result string buffer, return as much of the
2 ! quota record as will fit (zero filling the buffer).
2 !
2
2 CH$COPY (DQF$C_LENGTH, .Q_RECORD, 0,
2     .ABD[ABD$C_RES, ABD$W_COUNT],
2     .ABD[ABD$C_RES, ABD$W_TEXT] + ABD[ABD$C_RES, ABD$W_TEXT] + 1);
```

```
;  626          1612  2
;  627          1613  2 RETURN 1;
;  628          1614  2
;  629          1615  1 END;                          ! end of routine RET_QENTRY



                                        003C 00000          .ENTRY   RET_QENTRY, Save R2,R3,R4,R5                ; 1555
                        50        08  AC   D0 00002          MOVL     ABD, R0                                     ; 1599
                        02        1A  A0   B1 00006          CMPW     26(R0), #2
                                      0E   1F 0000A          BLSSU    1$
                        51        18  A0   9E 0000C          MOVAB    24(R0), R1                                  ; 1602
                        50            61   3C 00010          MOVZWL   (R1), R0
                              01 A140      9F 00013          PUSHAB   1(R1)[R0]
                        9E            20   B0 00017          MOVW     #32, @(SP)+
                        52        08  AC   D0 0001A 1$:      MOVL     ABD, R2                                     ; 1610
                        51        20  A2   9E 0001E          MOVAB    32(R2), R1                                  ; 1611
                        50            61   3C 00022          MOVZWL   (R1), R0
    22   A2         00        04    BC     20   2C 00025          MOVC5    #32, @Q_RECORD, #0, 34(R2), 1(R1)[R0]
                              01 A140         0002C
                        50            01   D0 0002F          MOVL     #1, R0                                      ; 1613
                                           04 00032          RET                                                  ; 1615

; Routine Size:  51 bytes,    Routine Base:  $CODE$ + 034F
```

```
1 GLOBAL ROUTINE CONN_QFILE (ABD, FIB) : L_NORM NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine causes the quota file for the volume set to be
1 !       connected and made active.
1 !
1 ! CALLING SEQUENCE:
1 !       CONN_QFILE (ARG1, ARG2)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: address of buffer descriptor vector
1 !       ARG2: address of user FIB
1 !
1 ! IMPLICIT INPUTS:
1 !       CLEANUP_FLAGS: cleanup action and status flags
1 !       CURRENT_RVN: RVN of currently selected volume
1 !       CURRENT_VCB: VCB of currently selected volume
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       PRIMARY_FCB: FCB created for quota file
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       directory searched, quota file accessed (FCB created, etc.)
1 !
1 !--
1
2 BEGIN
2
2 MAP
2       ABD               : REF BBLOCKVECTOR [,ABD$C_LENGTH],
2                                         ! buffer descriptor arg
2       FIB               : REF BBLOCK;   ! user FIB
2
2 LOCAL
2       FCB               : REF BBLOCK,   ! FCB of quota file
2       HEADER            : REF BBLOCK,   ! file header of quota file
2       BUFFER            : REF BBLOCK;   ! disk block buffer
2
2 BIND_COMMON;
2
2 EXTERNAL ROUTINE
2       REBLD_PRIM_FCB    : L_NORM NOVALUE, ! rebuild fcb from header
2       BUILD_EXT_FCBS    : L_NORM NOVALUE, ! build extension fcbs
2       ARBITRATE_ACCESS  : [ JSB_2ARGS,    ! arbitrate file access
2       SERIAL_FILE       : L_NORM,         ! serialize on given file
2       FIND              : L_NORM,         ! find file in directory
2       SWITCH_VOLUME     : L_NORM,         ! switch volume context
2       SEARCH_FCB        : L_NORM ADDRESSING_MODE (GENERAL), ! search FCB list
```

```
2               READ_HEADER     : L_NORM,           ! read file header
2               CREATE_FCB      : L_NORM;           ! create an FCB
2
2
2 ! Check caller privilege - must be "system".
2 !
2
2 IF NOT .CLEANUP_FLAGS[CLF_SYSPRV]
2 THEN ERR_EXIT (SS$_NOPRIV);
2
2 ! Find the quota file in the directory. The quota file must be located
2 ! RVN 1 if this is a volume set.
2 !
2
2 IF .CLEANUP_FLAGS[CLF_DIRECTORY]
2 THEN FIND (.ABD, .FIB, 0);
2 SWITCH_VOLUME (.FIB[FIB$W_FID_RVN]);
2 IF .CURRENT_RVN GTRU 1
2 THEN ERR_EXIT (SS$_BADQFILE);
2
2 ! Make sure the quota file is not already active.
2 !
2
2 IF .CURRENT_VCB[VCB$L_QUOTAFCB] NEQ 0
2 THEN ERR_EXIT (SS$_QFACTIVE);
2
2 ! Find the FCB, if any, and read the header.
2 !
2
2 SERIAL_FILE (FIB [FIB$W_FID]);
2
2 FCB = PRIMARY_FCB = SEARCH_FCB (FIB[FIB$W_FID]);
2
2 HEADER = READ_HEADER (FIB[FIB$W_FID]);
2
2 ! Create an FCB if none exists.
2 !
2
2 IF .FCB EQL 0
2 THEN
2     PRIMARY_FCB = FCB = CREATE_FCB (.HEADER)
2 ELSE
2     IF .FCB [FCB$V_STALE]
2     THEN
2         REBLD_PRIM_FCB (.FCB, .HEADER);
2
2 BUILD_EXT_FCBS (.HEADER);
2
2 ! Check the quota file for suitability (contiguous, file format, etc.)
2 !
2
2 IF NOT .HEADER[FH2$V_CONTIG]
2 OR .BBLOCK [HEADER[FH2$W_RECATTR], FAT$B_RTYPE] NEQ FAT$C_FIXED
2 OR .BBLOCK [HEADER[FH2$W_RECATTR], FAT$B_RATTRIB] NEQ 0
2 OR .BBLOCK [HEADER[FH2$W_RECATTR], FAT$W_RSIZE] NEQ DQF$C_LENGTH
2 THEN ERR_EXIT (SS$_BADQFILE);
2
```

RDB
V04

```
;  745      1730  2  ! Check access interlocks.
;  746      1731  2  !
;  747      1732  2
;  748      1733  2  IF NOT ARBITRATE_ACCESS (0, .FCB)
;  749      1734  2  THEN ERR_EXIT (SS$_ACCONFLICT);
;  750      1735  2
;  751      1736  2  ! Now hook up the quota file FCB.
;  752      1737  2  !
;  753      1738  2
;  754      1739  3  IF NOT KERNEL_CALL (MAKE_QFCB, .FCB)
;  755      1740  2  THEN ERR_EXIT (SS$_INSFMEM);                ! allocation failure on quota cache
;  756      1741  2
;  757      1742  1  END;                                        ! end of routine CONN_QFILE


                                                             .EXTRN   REBLD_PRIM_FCB, BUILD_EXT_FCBS
                                                             .EXTRN   ARBITRATE_ACCESS
                                                             .EXTRN   FIND, SEARCH_FCB
                                                             .EXTRN   READ_HEADER, CREATE_FCB

                                           000C 00000        .ENTRY   CONN_QFILE, Save R2,R3             ; 1616
                03          01       AA  E8 00002            BLBS     1(BASE), 1$                        ; 1680
                                     24  BF 00006            CHMU     #36                                ; 1681
                                         04 00008            RET
        0B                  6A       06  E1 00009 1$:        BBC      #6, (BASE), 2$                     ; 1687
                                     7E  D4 0000D            CLRL     -(SP)                              ; 1688
                7E          04       AC  7D 0000F            MOVQ     ABD, -(SP)                         ;
             0000G          CF       03  FB 00013            CALLS    #3, FIND                           ;
                50          08       AC  D0 00018 2$:        MOVL     FIB, R0                            ; 1689
                7E          08       A0  3C 0001C            MOVZWL   8(R0), -(SP)                       ;
             0000G          CF       01  FB 00020            CALLS    #1, SWITCH_VOLUME                  ;
                01          A0       AA  D1 00025            CMPL     -96(BASE), #1                      ; 1690
                                     76  1A 00029            BGTRU    6$                                 ;
                50          98       AA  D0 0002B            MOVL     -104(BASE), R0                     ; 1696
                            54       A0  D5 0002F            TSTL     84(R0)                             ;
                                     05  13 00032            BEQL     3$                                 ;
                          03CC       8F  BF 00034            CHMU     #972                               ; 1697
                                         04 00038            RET                                         ;
        7E        08        AC           04  C1 00039 3$:    ADDL3    #4, FIB, -(SP)                     ; 1702
             0000G          CF       01  FB 0003E            CALLS    #1, SERIAL_FILE                    ;
        7E        08        AC       04  C1 00043            ADDL3    #4, FIB, -(SP)                     ; 1704
         00000000G          00       01  FB 00048            CALLS    #1, SEARCH_FCB                     ;
                08          AA       50  D0 0004F            MOVL     R0, 8(BASE)                        ;
                            53       50  D0 00053            MOVL     R0, FCB                            ;
        7E        08        AC       04  C1 00056            ADDL3    #4, FIB, -(SP)                     ; 1706
             0000G          CF       01  FB 0005B            CALLS    #1, READ_HEADER                    ;
                            52       50  D0 00060            MOVL     R0, HEADER                         ;
                                     53  D5 00063            TSTL     FCB                                ; 1711
                                     10  12 00065            BNEQ     4$                                 ;
                                     52  DD 00067            PUSHL    HEADER                             ; 1713
             0000G          CF       01  FB 00069            CALLS    #1, CREATE_FCB                     ;
                            53       50  D0 0006E            MOVL     R0, FCB                            ;
                08          AA       53  D0 00071            MOVL     FCB, 8(BASE)                       ;
                                     0D  11 00075            BRB      5$                                 ;
                09          23       A3  E9 00077 4$:        BLBC     35(FCB), 5$                        ; 1715
                                     52  DD 0007B            PUSHL    HEADER                             ; 1717
```

```
                                 53  DD 0007D              PUSHL    FCB                                   ;
        0000G  CF                02  FB 0007F              CALLS    #2, REBLD_PRIM_FCB                    ;
                                 52  DD 00084  5$:         PUSHL    HEADER                                ; 1719
        0000G  CF                01  FB 00086              CALLS    #1, BUILD_EXT_FCBS                    ;
                          34     A2  95 0008B              TSTB     52(HEADER)                            ; 1724
                                 11  18 0008E              BGEQ     6$                                    ;
               01         14     A2  91 00090              CMPB     20(HEADER), #1                        ; 1725
                                 0B  12 00094              BNEQ     6$                                    ;
                          15     A2  95 00096              TSTB     21(HEADER)                            ; 1726
                                 06  12 00099              BNEQ     6$                                    ;
               20         16     A2  B1 0009B              CMPW     22(HEADER), #32                       ; 1727
                                 05  13 0009F              BEQL     7$                                    ;
                        03BC     8F  BF 000A1  6$:         CHMU     #956                                  ; 1728
                                     04 000A5              RET                                            ;
               51                53  D0 000A6  7$:         MOVL     FCB, R1                               ; 1733
                                 50  D4 000A9              CLRL     R0                                    ;
                              0000G  30 000AB              BSBW     ARBITRATE_ACCESS                      ;
               05                50  E8 000AE              BLBS     R0, 8$                                ;
                        0800     8F  BF 000B1              CHMU     #2048                                 ; 1734
                                     04 000B5              RET                                            ;
                                 53  DD 000B6  8$:         PUSHL    FCB                                   ; 1739
        0000V  CF                01  FB 000B8              CALLS    #1, MAKE_QFCB                         ;
               04                50  E8 000BD              BLBS     R0, 9$                                ;
                        0124     8F  BF 000C0              CHMU     #292                                  ; 1740
                                     04 000C4  9$:         RET                                            ; 1742

; Routine Size:  197 bytes,    Routine Base:  $CODE$ + 0382
```

```
1 GLOBAL ROUTINE MAKE_QFCB (FCB) : L_NORM =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine hooks up the specified FCB to be the FCB for the
1 !       volume (set) quota file. This routine must be called in kernel mode.
1 !
1 ! CALLING SEQUENCE:
1 !       MAKE_QFCB (ARG1)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: address of FCB to hook up
1 !
1 ! IMPLICIT INPUTS:
1 !       CURRENT_VCB: VCB of volume
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       1 if successful
1 !       0 if allocation failure on cache block
1 !
1 ! SIDE EFFECTS:
1 !       quota file FCB hooked into FCB list and quota pointer
1 !
1 !--
1
2 BEGIN
2
2 MAP
2       FCB             : REF BBLOCK;   ! FCB to hook up
2
2 LOCAL
2       QUOTA_CACHE     : REF BBLOCK,   ! quota cache block allocated
2       ACB             : REF BBLOCK;   ! AST control block within quota block
2
2 BIND_COMMON;
2
2 EXTERNAL
2       SCH$GL_SWPPID   : ADDRESSING_MODE (GENERAL);
2                                       ! PID of swapper process
2
2 EXTERNAL ROUTINE
2       ALLOCATE        : L_NORM ADDRESSING_MODE (GENERAL), ! allocate system dynamic memory
2       CACHE_LOCK      : L_NORM,       ! get special cache lock
2       XQP$UNLOCK_QUOTA : ADDRESSING_MODE (GENERAL);
2                                       ! release lock with value block
2
2
2 ! Allocate the cache block and link it to the VCB.
2 !
```

```
;  816      1800  2
;  817      1801  2 QUOTA_CACHE = ALLOCATE (MAXU (.CURRENT_VCB[VCB$W_QUOSIZE], 1) * VCA$C_QUOLENGTH
;  818      1802  2                         + $BYTEOFFSET (VCA$L_QUOLIST), CACHE_TYPE);
;  819      1803  2 IF .QUOTA_CACHE EQL 0
;  820      1804  2 THEN RETURN 0;
;  821      1805  2 QUOTA_CACHE[VCA$W_QUOSIZE] = MAXU (.CURRENT_VCB[VCB$W_QUOSIZE], 1);
;  822      1806  2 CURRENT_VCB[VCB$L_QUOCACHE] = .QUOTA_CACHE;
;  823      1807  2
;  824      1808  2 ! Initialize the AST control blocks in the quota cache header. One is
;  825      1809  2 ! used to post blocking AST's to the swapper to release cache entries.
;  826      1810  2 ! The other is used to trip the cache flush process to flush the entire
;  827      1811  2 ! cache.
;  828      1812  2 !
;  829      1813  2
;  830      1814  2 ACB = QUOTA_CACHE[VCA$B_QUOACB];
;  831      1815  2 ACB[ACB$B_RMOD] = PSL$C_KERNEL + ACB$M_NODELETE;
;  832      1816  2 ACB[ACB$L_PID] = .SCH$GL_SWPPID;
;  833      1817  2 ACB[ACB$L_AST] = XQP$UNLOCK_QUOTA;
;  834      1818  2 ACB = QUOTA_CACHE[VCA$B_QUOFLUSHACB];
;  835      1819  2 ACB[ACB$B_RMOD] = PSL$C_KERNEL + ACB$M_NODELETE;
;  836      1820  2
;  837      1821  2 ! Bump up the access counts in the FCB to show an accessed file.
;  838      1822  2 ! Lock it against truncates.
;  839      1823  2 !
;  840      1824  2
;  841      1825  2 FCB[FCB$W_REFCNT] = .FCB[FCB$W_REFCNT] + 1;
;  842      1826  2 FCB[FCB$W_ACNT] = .FCB[FCB$W_ACNT] + 1;
;  843      1827  2 FCB[FCB$W_TCNT] = .FCB[FCB$W_TCNT] + 1;
;  844      1828  2
;  845      1829  2 ! If the quota file is already write accessed, take out the cache lock
;  846      1830  2 ! on the write access to prevent use of the cache.
;  847      1831  2 !
;  848      1832  2
;  849      1833  2 IF .FCB[FCB$W_WCNT] NEQ 0
;  850      1834  2 AND .BBLOCK [CURRENT_UCB[UCB$L_DEVCHAR2], DEV$V_CLU]
;  851      1835  2 AND .FCB[FCB$L_CACHELKID] EQL 0
;  852      1836  2 THEN CACHE_LOCK (.FCB[FCB$L_LOCKBASIS], FCB[FCB$L_CACHELKID], 2);
;  853      1837  2
;  854      1838  2 ! Finally enter the quota file pointer in the VCB.
;  855      1839  2 !
;  856      1840  2
;  857      1841  2 CURRENT_VCB[VCB$L_QUOTAFCB] = .FCB;
;  858      1842  2
;  859      1843  2 CLEANUP_FLAGS[CLF_DEACCQFILE] = 1;
;  860      1844  2
;  861      1845  2 RETURN 1;
;  862      1846  2
;  863      1847  1 END;                                         ! end of routine MAKE_QFCB


                                                          .EXTRN  SCH$GL_SWPPID, ALLOCATE
                                                          .EXTRN  CACHE_LOCK, XQP$UNLOCK_QUOTA

                                        0000 00000        .ENTRY  MAKE_QFCB, Save nothing                          ; 1743
                                      06  DD 00002        PUSHL   #6                                              ; 1801
                      50       98     AA  D0 00004        MOVL    -104(BASE), R0                                   ;
```

```
                    50       60   A0  3C 00008          MOVZWL  96(R0), R0                                    :
                                  03  12 0000C          BNEQ    1$                                            :
                    50            01  D0 0000E          MOVL    #1, R0                                        :
                    50            1C  C4 00011 1$:      MULL2   #28, R0                                       :
                             44   A0  9F 00014          PUSHAB  68(R0)                                        : 1802
          00000000G 00            02  FB 00017          CALLS   #2, ALLOCATE                                  :
                    51            50  D0 0001E          MOVL    R0, QUOTA_CACHE                               :
                                  03  12 00021          BNEQ    2$                                            : 1803
                             0081  31 00023          BRW     5$                                            :
                    50       98   AA  D0 00026 2$:      MOVL    -104(BASE), R0                                : 1805
                    50       60   A0  3C 0002A          MOVZWL  96(R0), R0                                    :
                                  03  12 0002E          BNEQ    3$                                            :
                    50            01  D0 00030          MOVL    #1, R0                                        :
                    61            50  B0 00033 3$:      MOVW    R0, (QUOTA_CACHE)                             :
                    50       98   AA  D0 00036          MOVL    -104(BASE), R0                                : 1806
               5C   A0            51  D0 0003A          MOVL    QUOTA_CACHE, 92(R0)                           :
                    50       0C   A1  9E 0003E          MOVAB   12(R1), ACB                                   : 1814
               0B   A0            20  90 00042          MOVB    #32, 11(ACB)                                  : 1815
               0C   A0 00000000G  00  D0 00046          MOVL    SCH$GL_SWPPID, 12(ACB)                        : 1816
               10   A0 00000000G  00  9E 0004E          MOVAB   XQP$UNLOCK_QUOTA, 16(ACB)                     : 1817
                    50       28   A1  9E 00056          MOVAB   40(R1), ACB                                   : 1818
               0B   A0            20  90 0005A          MOVB    #32, 11(ACB)                                  : 1819
                    50       04   AC  D0 0005E          MOVL    FCB, R0                                       : 1825
                             18   A0  B6 00062          INCW    24(R0)                                        :
                    50       04   AC  D0 00065          MOVL    FCB, R0                                       : 1826
                             1A   A0  B6 00069          INCW    26(R0)                                        :
                    50       04   AC  D0 0006C          MOVL    FCB, R0                                       : 1827
                             20   A0  B6 00070          INCW    32(R0)                                        :
                    50       04   AC  D0 00073          MOVL    FCB, R0                                       : 1833
                             1C   A0  B5 00077          TSTW    28(R0)                                        :
                                  1A  13 0007A          BEQL    4$                                            :
                    51       94   AA  D0 0007C          MOVL    -108(BASE), R1                                : 1834
                    12       3C   A1  E9 00080          BLBC    60(R1), 4$                                    : 1835
                             54   A0  D5 00084          TSTL    84(R0)                                        : 1835
                                  0D  12 00087          BNEQ    4$                                            :
                                  02  DD 00089          PUSHL   #2                                            : 1836
                             54   A0  9F 0008B          PUSHAB  84(R0)                                        :
                             4C   A0  DD 0008E          PUSHL   76(R0)                                        :
              0000G CF            03  FB 00091          CALLS   #3, CACHE_LOCK                                :
                    50       98   AA  D0 00096 4$:      MOVL    -104(BASE), R0                                : 1841
               54   A0       04   AC  D0 0009A          MOVL    FCB, 84(R0)                                   :
               03   AA            02  88 0009F          BISB2   #2, 3(BASE)                                   : 1843
                    50            01  D0 000A3          MOVL    #1, R0                                        : 1845
                                      04 000A6          RET                                                   :
                                  50  D4 000A7 5$:      CLRL    R0                                            : 1847
                                      04 000A9          RET                                                   :

; Routine Size:  170 bytes,    Routine Base:  $CODE$ + 0447


;  864          1848  1
;  865          1849  1 END
;  866          1850  0 ELUDOM
;                                   PSECT SUMMARY

;       Name                    Bytes                         Attributes

;  $CODE$                             1265  NOVEC,NOWRT, RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                              Library Statistics

;                                            -------- Symbols --------       Pages        Processing
;       File                                 Total    Loaded   Percent       Mapped       Time

;  _$255$DUA28:[SYSLIB]LIB.L32;1             18619      103         0        1000           00:01.9



;                                    COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:QUOTAUTIL/OBJ=OBJ$:QUOTAUTIL MSRC$:QUOTAUTIL/UPDATE=(ENH$:QUOTAUTIL)

; Size:           1265 code + 0 data bytes
; Run Time:          01:03.6
; Elapsed Time:      01:58.4
; Lines/CPU Min:      1746
; Lexemes/CPU-Min: 57359
; Memory Used:  321 pages
; Compilation Complete
```

0171 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

PMS
LIS

QUOTAUTIL
LIS

MAKPTR
LIS

MATCHNAME
LIS

MPWIND
LIS

LOCKDB
LIS

LOCKERS
LIS

MAKACC
LIS

PARSNM
LIS

MAPVBN
LIS

IODONE
LIS

LOCKDN
LIS

MODIFY
LIS

MOUNT
LIS

NXTHDR
LIS

MAKNMB
LIS

MAKSTR
LIS

0172 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY